

**Amendment to Specification**

Please replace paragraph 6 with the following amended paragraph:

[0006] Fig. 1 represents an illustration of a prior art autonomic system 100 that depicts the foregoing shortcomings of the prior art autonomic system. More specifically, in the prior art autonomic software system 100, an application is executed in a run time environment 102 via a profiling agent 104. The profiling agent 104 profiles the executing application 106 to generate a profile 110. The profile is transmitted to a prior art controller 114 and phase shift detector 124. The controller 114 controls the timing of when the profile 110 is collected via a control signal 108. It is noted that a plurality of profiles may be collected by the profiling agent 104, as directed by the controller 114 via the control signal 108. Generally, the controller 114 comprises a behavior adjustor 116 that adjusts the behavior of the executing application 106 by applying one or more actions 112 to the executing application. It is noted that the behavior adjustor 116 may apply no actions to the executing application as will be discussed below. The behavior adjustor 116 comprises an analyzer 118, selector 120 and applicator 122. The analyzer 118 receives and analyzes the profile 110. More specifically, the analyzer 118 receives raw data via profile 110 and transforms the raw data into a set of data that is utilizable by the controller 114. The set of data may include a sequence of basic blocks, pairs of method IDs and associated execution times, sequences of instruction cache misses, etcetera. For example, the analyzer 118 may analyze raw data, such as a plurality of method identifiers, and transform the raw data into canonical form, such as a weighted set of method identifiers where the weight of each method identifier equals the number of times the method identifier appeared in the raw data. The selector 120 receives the set of data from the analyzer 118 and selects one or more actions to be applied 112 to the executing application 106 that may help to improve the behavior of the executing application. In its determination, the selector 120 may use a cost-benefit model to determine if the cost of optimizing a method, represented by a method identifier for example, is less than the expected optimization benefit from a subsequent execution of the method, i.e., whether optimizing the method would improve the application's performance. The selector 120 then transmits the selected one or more selected actions 112 ~~actions~~ to the applicator 122, which applies the one or more actions 112 to the executing application 106. It is noted, that the selector

may select no actions based on the cost-benefit model, in which case the applicator 122 will not apply any actions. An article by Arnold, Mathew, *et al.* ("Adaptive Optimization in the Jalapeno JVM." Proceedings of the ACM Conference on Object-Oriented Programming Systems, Languages, and Applications Oct. 2000), discloses the foregoing example. During a subsequent predetermined time interval, the profiling agent 104 ~~102~~ may collect another profile 110 of the executing application 106 as controlled via the control signal 108. As is clear from the prior art autonomic system 100 of Fig. 1, the shortcoming of the autonomic system 100 is that neither the controller 114 nor the phase shift detector 124 has the benefit of utilizing the profile 110 obtained from the executing application that accounts for the one or more actions 112 applied to the executing application 106, after the profile 110 is collected from the executing application 106, but before the subsequent profile is collected from the executing application 106.